This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/808,781	03/15/2001	Dov Dori	SIA-001.01	5493

25181 7590 08/24/2004

FOLEY HOAG, LLP
PATENT GROUP, WORLD TRADE CENTER WEST
155 SEAPORT BLVD
BOSTON, MA 02110

EXAMINER	
PROCTOR, JASON SCOTT	

ART UNIT	PAPER NUMBER
2123	

DATE MAILED: 08/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/808,781	DORI, DOV	
	Examiner	Art Unit	
	Jason Proctor	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-34 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-34 is/are rejected.

7)☒ Claim(s) 2,3,8,13,17 and 33 is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 27 August 2001 is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Oath/Declaration

1. The examiner notes that a Declaration and Power of Attorney was filed and received with the appropriate fee, however the document is missing. The office is investigating this matter, however the applicant may wish to resubmit the missing document.

Priority

2. Applicant's claim for domestic priority under 35 U.S.C. §119(e) is acknowledged. Applicant's domestic priority under 35 U.S.C. §119(e) has been granted as May 4, 2000.

Drawings

3. Replacement drawings were received on 8/27/2001. These drawings are accepted for examination purposes only.

4. The drawings are objected to for the following informalities:

5. Figure 31 should be designated "FIG. 31A" to indicate its status as a partial view in association with Figure 31B. Additionally, the boxes labeled "FIG. 31A" and "FIG. 31B" should be removed to avoid confusion. See MPEP 608.02(u).

6. The figure of sheet 35 is not labeled with a figure number. The specification (page 44, line 21 – page 45, line 23) makes reference to Figure 38, which is presumed to be the figure of sheet 35. See MPEP 608.02(u).

Appropriate correction is required.

Specification

7. 35 U.S.C. 112, first paragraph, requires the specification to be written in "full, clear, concise, and exact terms." The specification is replete with terms which are not clear, concise and exact. The specification should be revised carefully in order to comply with 35 U.S.C. 112, first paragraph. Examples of some unclear, inexact or verbose terms used in the specification are:

8. "In addition, multimedia artifacts can be switched on or off along with generation or destruction of objects or" (page 42, lines 15-17).

9. "15°C" (page 25, lines 23-26) does not correspond to "15 deg" (Figure 25).

10. "The default is s, so if no unit is specified, it is seconds," (page 26, lines 29-30) and "Additionally, as indicated by the bolding of the "single" 108 state's border, the designer has specified that a "Person" 106 object initially assumes the "single" 108 state" (page 8, lines 6-9), while the "s" element of Figure 26 has a normal, non-bold border.

11. The reference to Figure 20, element 240 is not found in Figure 20 and presumed to be Figure 20, element 340 (page 23, lines 12-20).

12. The reference to Figure 20, element 350 is "Signature Set", not "Signed" as it appears (page 23, lines 12-20).

13. The reference to Figure 20, element 356 is "Notary Seal", not "Notary Sealed" as it appears (page 23, lines 12-20).

14. The use of the term "hyper-graph" (page 18, lines 24-27) is unknown and conflicts with the known definition of "hypergraph". A hypergraph is a graph in which generalized edges (called hyperedges) may connect more than two nodes (Eric W. Weisstein. "Hypergraph." From MathWorld--A Wolfram Web Resource).

15. The specification discloses that the tool formats labeled elements in bold while reserved phrases appear in a normal, non-bolded font (page 9, lines 1-3). Appendix B (page 1, paragraph 9), discloses that non-bold Arial words in the Production Rules denote reserved terminals. According to Appendix B, page 4, the word "affects" in an "Effect-sentence" is a reserved terminal and, according to the specification as cited, should be displayed in a normal, non-bolded font, however Figure 12, a screenshot of the tool (page 4, lines 9-10), shows the reserved terminal word "affects" in bold font.

16. Regarding Figure 23, a listing of formal text which corresponds to the OPM diagram of Figures 21A and 21B (page 4, lines 13-14), states "Attribute instantiates Value" while Figures 21A and 21B show, in the language of OPM model elements listed in Appendix A, that "Attribute instantiates Nalue", not Value. Additionally, Figure 23 states "Ellipse, Rectangle, and Rountangle are closed Shapes," while Figures 21A and 21B clearly show that Ellipse, Rectangle, and Rountangle are Shapes, not closed Shapes. Regarding "Attribute instantiates Nalue" and "Ellipse, Rectangle, and Rountangle are Shapes", the specification discloses that the text description and graphic diagram are equivalent in informational content (page 6, lines 1-3) which would require an oversight or error in one to be duplicated in the other.

Appropriate correction is required.

Claim Objections

17. Claims 2 and 17 are objected to because of the following informalities: The claim language is unclear. Additional punctuation in line 2 would significantly clarify the meaning of the language. Examiner suggests "The method of claim 1 (16), wherein the graphic elements correspond to a notation, modeling objects and processes as independent elements." See MPEP 1.75(a). The examiner presumes the claim is to be punctuated as suggested for the remainder of this prosecution. Appropriate correction is required.

18. Claim 3 is objected to because of the following informalities: The term "the notation" in line 1 lacks proper antecedent basis, however it is presumed for the remainder of the prosecution that Claim 3 depends upon the method of Claim 2 rather than the method of Claim 1, thus providing proper antecedent basis. Appropriate correction is required.

19. Claim 8 is objected to because of the following informalities: The phrase "wherein the generating" in line 1 contains grammatical errors. The examiner presumes "wherein generating" for the remainder of this action. Appropriate correction is required.

20. Claim 33 is objected to because it ends with improper punctuation. See MPEP 608.01(m). The examiner presumes proper punctuation for the remainder of the prosecution. Appropriate correction is required.

Claim Rejections - 35 USC § 112

21. The following is a quotation of the first paragraph of 35 U.S.C. §112:

> **The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.**

22. Claims 4, 5, 13, 19-21, and 30-32 are rejected under 35 U.S.C. §112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

23. Regarding all claims rejected under 35 U.S.C. §112, first paragraph, said claims recite limitations involving the context free grammar and/or production rules based thereon or depend claims that recite such limitations. Appendix B contains Production Rules for a Context Free Grammar (CFG) which are incomplete and do not correspond with the drawings or specification.

24. As an example of the Production Rules failing to correspond to the specification and drawings, the "Diagram-id" rule (Appendix B, page 6) resolves to the terminal symbols represented by "SD || (/dash integer)*", such as "SD-1" or "SD-999" as example of the examiner's interpretation. Figure 12, element 230 depicts "SD" and "SD1" as Diagram-id symbols within a "process-detailing-sentence", as interpreted by the examiner. Further, there is no production rule listed in Appendix B which properly corresponds to element 230.

25. As another example, Figures 1-9, elements 202 and 208 relate to an initial or default state for which there is no production rule in Appendix B.

26. As another example, Figure 2, element 204 is interpreted as "Exhibition-sentence", however the predicate of the sentence does not properly correspond to any portion of the right hand side of the production rule for an "Exhibition-sentence".

27. As an example of the incompleteness of the context free grammar production rules of Appendix A, there is no production rule for the non-terminal symbols "Object-detailing-sentence" and "State-detailing-sentence" which resolves said non-terminals into one or more terminal symbols.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

28. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> **The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.**

29. Claims 3, 5, 13, 18, 20, 21, 25, 31, and 32 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

30. Regarding Claims 3, 18, and 31, it is unclear how broad the definition of "Object Process Methodology" can be correctly interpreted. See Claim Interpretation below.

31. Regarding Claims 5 and 20, it is unclear how a context-free grammar expression can be generated from a context-free production rule such that it is consistent with a natural language. For example, it is unclear how the context-free production rules of

Appendix B would produce expressions with properly conjugated English verbs while retaining the property of being context-free. See Claim Interpretation below.

32. Claim 21 is rejected by virtue of its dependence on Claim 20.

33. Regarding Claims 13 and 32, it is unclear how a textual description can be generated using production rules of a context-free grammar for a natural language. The known definition for a natural language has no requirement to be context-free and it is unclear how production rules for a context-free grammar can produce a potentially context-sensitive language such as a natural language. Natural language: a naturally evolved human language as opposed to a created language such as a computer language (Encarta World English Dictionary, North American Edition). See Claim Interpretation below.

34. Regarding Claim 25, the claim limitations refer to generating a textual description though the limitations of Claim 16, on which Claim 25 depends, state that the textual description is received as input. It is unclear what is meant by generating textual input that is received as input. See Claim Interpretation below.

Claim Interpretation

35. The broadest reasonable interpretation of the claims has been afforded the instant claimed invention. However, due to the rejections under 35 U.S.C. §112 above, the following claim interpretations have been made. The interpretations have been drawn from the teachings held within the specification as filed and that of teachings in the prior art.

36. Regarding Claims 3, 18, and 31, "Object-Process Methodology" has been interpreted to mean "graphical notation that models objects and processes as independent elements" (page 2, lines 11-13).

37. Regarding Claim 5, the limitation "wherein the production rules comprises production rules consistent with a natural language" has been interpreted as "wherein the production rules generate expressions using vocabulary from a natural language," (Appendix B).

38. Regarding Claim 13, the limitation "generating the textual description comprises generating using production rules of a context-free grammar for the second natural language" has been interpreted as "generating the textual description comprises generating using production rules of a context-free grammar using vocabulary from the second natural language," (Appendix B).

39. Regarding Claim 20, the limitation "wherein the context-free grammar comprises a grammar having production rules consistent with a natural language" has been interpreted as "wherein the context-free grammar comprises a grammar using vocabulary consistent with a natural language," (Appendix B).

40. Regarding Claim 25, the limitation is interpreted as "wherein generating the diagram comprises generating in batch mode," (page 7, lines 10-14).

41. Regarding Claim 32, the limitation "the context-free grammar being consistent with the syntax of the second language" is interpreted as "the context-free grammar using vocabulary of the second language," (Appendix B).

42. Regarding Claims 4, 5, 13, 19-21, and 30-32, "context free grammar" and "production rules" have been interpreted as though the Production Rules of Appendix B were complete and the drawings, specification, and Appendix B were all in accordance with each other.

Claim Rejections - 35 USC § 102

43. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> **A person shall be entitled to a patent unless –**
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

44. Claim 1-11, 14, and 33 are rejected under 35 U.S.C. 102(b) as being anticipated by Marmelstein.

45. Regarding Claim 1, Marmelstein teaches a program code generation tool which

receives input from the user that specifying a program in the form of a graphical flowchart (column 2, lines 58-63),

presents a textual representation of the flowchart (column 2, line 63 - column 3, line 4),

is updated to accurately correspond to the flowchart (column 3, lines 38-40; column 10, lines 54-61),

represents objects using rectangles (column 17, lines 52-65; figure 5),

represents operations associated with the objects using circles (column 17, lines 59-65; figure 5; column 22, lines 25-41; figure 6),

produces a program code textual description of the flowchart (column 2, lines 59-63; figures 15-25; column 11, lines 10-36), and

can produce a graphical representation of a flowchart by reading the corresponding program code textual description of the model (column 3, lines 9-13; column 18, lines 36-49; column 27, lines 8-13).

46. Regarding Claim 2, Marmelstein teaches a program code generation tool wherein the tool models objects as rectangles in the APEX Editor (column 17, lines 52-65; figure 5) and operations associated with the objects are independently represented as circles in the REM-Net Editor (column 17, lines 59-65; figure 5; column 22, lines 25-41; figure 6).

47. Regarding Claim 3, Marmelstein teaches a program code generation tool wherein the tool models objects as rectangles in the APEX Editor (column 17, lines 52-65; figure 5) and operations associated with the objects are independently represented as circles in the REM-Net Editor (column 17, lines 59-65; figure 5; column 22, lines 25-41; figure 6).

48. Regarding Claim 4, Marmelstein teaches a program code generation tool wherein the tool produces an ADA programming language textual description of a flowchart (column 2, lines 59-63; figures 15-25; column 11, lines 10-36). The use of context-free grammars to define programming language syntax is well known, indeed this was the first application of context-free grammars (See Sebesta, pages 109-110).

A context-free grammar for ADA is known (See The Ada 95 Reference Manual, Annex P). Thus generating an ADA programming language textual description is deemed equivalent to generating a context-free grammar expression.

49. Regarding Claim 5, Marmelstein teaches a program code generation tool wherein the tool produces an ADA programming language textual description of a flowchart (column2, lines 59-63; figures 15-25; column 11, lines 10-36). ADA is a high level programming language (See Microsoft Computer Dictionary, Fifth Edition). Further, statements in a high level programming language generally use keywords similar to English (See Microsoft Computer Dictionary, Fifth Edition). Thus Marmelstein teaches generating a textual description of a flowchart according to a context-free grammar that uses vocabulary from English, a natural language.

50. Regarding Claim 6, Marmelstein teaches displaying both the flowchart and the corresponding textual description simultaneously (figures 5-8; column 2, line 63 – column 3, line 9; column 4, lines 5-8).

51. Regarding Claim 7, Marmelstein teaches a user giving input to the program code generation tool (column 6, line 64 – column 7, line 6)

52. Regarding Claim 8, Marmelstein teaches a program code generation tool where the textual description corresponding to the flowchart is synchronously updated to reflect changes the user enacts (column 2, line 59 – column 3, line 4; column 3, lines 38-40; column 10, lines 54-61)

53. Regarding Claim 9, Marmelstein teaches a program code generation tool where the textual description corresponding to the flowchart can be generated in its entirety in

response to a user's command (column 11, lines 10-15; column 11, lines 44-48; column 18, lines 53-57; figure 5).

54. Regarding Claim 10, Marmelstein teaches a program code generation tool where the user can specify a level of detail to depict by giving input into the APEX editor (column 6, lines 36-46) or the DS editor (column 6, lines 47-62). Also, the APEX editor permits the user to represent the program's top level package structure (column 17, lines 52-69) while certain user input will cause the program to depict a higher or lower level of detail (column 17, line 59 – column 18, line 2).

55. Regarding Claim 11, Marmelstein teaches a program code generation tool where the Data Structure Editor allows the user to provide input that causes the tool to display greater detail regarding the ADA program code textual description of the flowchart. The Data Structure Editor displays certain ADA program code such as data structures, subprograms, and variables in the scope of the current data structure (column 24, lines 43-51; column 25, lines 3-16).

56. Regarding Claim 14, Marmelstein teaches a program code generation tool which generates program code to implement the flowchart (column 2, lines 59-63; figures 15-25; column 11, lines 10-36).

57. Regarding Claim 33, Marmelstein teaches a computer program product which

receives input from the user that specifying a program in the form of a graphical flowchart (column 2, lines 58-63),

presents a textual representation of the flowchart (column 2, line 63 - column 3, line 4),

which is updated to accurately correspond to the flowchart (column 3, lines 38-40; column 10, lines 54-61),

the tool represents objects using rectangles (column 17, lines 52-65; figure 5),

the tool represents operations associated with the objects using circles (column 17, lines 59-65; figure 5; column 22, lines 25-41; figure 6),

the tool produces a program code textual description of the flowchart (column 2, lines 59-63; figures 15-25; column 11, lines 10-36), and

can produce a graphical representation of a flowchart by reading the corresponding program code textual description of the model (column 3, lines 9-13; column 18, lines 36-49; column 27, lines 8-13).

58. Claims 16-29 and 34 are rejected under 35 U.S.C. 102(b) as being anticipated by Kossiakoff.

59. Regarding Claim 16, Kossiakoff teaches a method of modeling through the use of a computer program wherein

text input is received from a user to specify an electronic circuit diagram (column 4, lines 59-62; column 24, lines 42-50),

the circuit diagram being composed of graphic elements (column 3, lines 5-10; column 4, lines 42-52),

where polygons represent data circuit elements (figure 3; column 5, lines 37-38; column 3, lines 26-35), the objects of the model, and

solid and dashed lines represent the flow of data and control signals (column 6, lines 28-35), the processes of the model.

Further, the invention disclosed in Kossiakoff creates program code corresponding to the electronic circuit diagram, thereby modeling the operation of the electronic circuit, so that the user may detect and correct errors (column 4, line 63 – column 5, line 5).

60. Regarding Claim 17, Kossiakoff teaches a method of modeling through the use of a computer program where the data flow circuit elements represent hardware building blocks (column 7, lines 35-43), thereby modeling objects in the circuit. In the diagram, solid lines represent the flow of data while dashed lines represent control signals (column 6, lines 28-35), thereby modeling processes within the circuit.

61. Regarding Claim 18, Kossiakoff teaches a method of modeling through the use of a computer program where objects and processes are graphically represented as individual elements (figures 2-3; column 6, lines 28-35; column 7, lines 35-43).

62. Regarding Claim 19, Kossiakoff teaches a method of modeling through the use of a computer program that receives input from a user specifying an electronic circuit diagram (column 4, lines 59-62; column 24, lines 42-50). It is deemed inherent that the text input forming the computer-human interface consists of a context-free grammar. The understanding and use of natural languages is a much harder problem for computers than understanding computer languages (See TECHNICAL, "natural language").

63. Regarding Claim 20, Kossiakoff teaches a method of modeling through the use of a computer program that receives input from a user specifying an electronic circuit diagram (column 4, lines 59-62; column 24, lines 42-50). It is deemed inherent that the text input forming the computer-human interface consists of a context-free grammar. It is deemed inherent that the design of a context-free grammar for use as a computer-human interface would consist of vocabulary from the natural language of the intended user.

64. Regarding Claim 21, Kossiakoff teaches a method of modeling through the use of a computer program which receives input from a user specifying an electronic circuit diagram (column 4, lines 59-62; column 24, lines 42-50). It is deemed inherent that the method disclosed requires parsing the received input in according to the context-free grammar recognized by the program.

65. Regarding Claim 22, Kossiakoff teaches that an object of the invention is to provide an alpha-numerically documented representation of the process to be modeled such that it is clearly understandable by an engineer, who represents the domain expert (column 3, lines 53-59).

66. Regarding Claim 23, Kossiakoff teaches that the user provides input to specify the electronic circuit diagram (column 4, lines 59-62; column 24, lines 42-50).

67. Regarding Claim 24, Kossiakoff teaches that the user may use a light pen to interactively create the diagram (column 12, lines 16-27).

68. Regarding Claim 26, Kossiakoff teaches that the circuit diagram may comprise "program blocks", which are themselves circuit diagrams (figures 6-7; column 22, lines

1-16; column 22, lines 32-39). The user manipulates these program blocks through the use of an Integration Editor, while the circuit diagrams represented by the program blocks are manipulated through the use of a separate editor (column 21, lines 56-69; column 12, lines 44-53). It is deemed inherent that the selection of either the data circuit editor or the Integration Editor is the result of user input.

69. Regarding Claim 27, Kossiakoff teaches that the circuit diagram may comprise "program blocks", which are themselves circuit diagrams (figures 6-7; column 22, lines 1-16; column 22, lines 32-39). The user manipulates these program blocks through the use of an Integration Editor, while the circuit diagrams represented by the program blocks are manipulated through the use of a separate editor (column 21, lines 56-69; column 12, lines 44-53). It is deemed inherent that the selection of either the data circuit editor or the Integration Editor is the result of user input. These editors display different levels of detail in the diagram (column 5, lines 55-60; figures 7-8).

70. Regarding Claim 28, Kossiakoff teaches that the program automatically generates software instructions to implement the model (column 3, lines 26-35; column 4, line 63 – column 5, line 5; column 22, lines 32-39).

71. Regarding Claim 29, Kossiakoff teaches that the circuit diagram is converted into computer code and tested with sample inputs and outputs. The goal is to correct errors or omissions by the designer (column 4, line 66 – column 5, line 5). It is deemed that this constitutes a simulation of the modeled system.

72. Regarding Claim 34, Kossiakoff discloses a computer program product wherein

text input is received from a user to specify an electronic circuit diagram (column 4, lines 59-62; column 24, lines 42-50),

the circuit diagram being composed of graphic elements (column 3, lines 5-10; column 4, lines 42-52),

where polygons represent data circuit elements (figure 3; column 5, lines 37-38; column 3, lines 26-35), the objects of the model, and

solid and dashed lines represent the flow of data and control signals (column 6, lines 28-35), the processes of the model.

Further, the invention disclosed in Kossiakoff creates program code corresponding to the electronic circuit diagram, thereby modeling the operation of the electronic circuit, so that the user may detect and correct errors (column 4, line 63 – column 5, line 5).

Claim Rejections - 35 USC § 103

73. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> **(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.**

74. Claims 12 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Marmelstein as applied to claim 1 above, and further in view of Fukumochi et al.

75. Regarding Claim 12, Marmelstein does not teach the translation of labeled diagram elements from one natural language to another, however Fukumochi et al.

discloses an automatic translating machine which makes use of parse trees (Fukumochi et al., column 3, line 58 – column 4, line 13). The combination of the translation device disclosed by Fukumochi et al. with the programming code generation tool disclosed by Marmelstein could be realized as a feature initiated by user command to translate the labels of the various graphical elements. It is deemed that the advantages of a translating the labels of a flowchart representing a proposed program design into a second language are inherent in view of the various natural languages spoken by software developers. It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to incorporate the translation ability of Fukumochi et al. with the automatic diagramming and code generation ability of Marmelstein to improve the usability of the code generation tool.

76. Regarding Claim 13, the limitations of Claim 12 are rejected as above, and further Fukumochi et al. discloses the use of parse trees (column 3, line 58 – column 4, line 13) which are used to represent the syntactic structure of context-free grammars (Sebesta, 113). Additionally, Fukumochi et al. discloses the use of production rules which conform to a subset of grammatical rules for a natural language (column 2, Table 2; column 2, lines 28-35; column 6, lines 1-21).

77. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Marmelstein as applied to claim 1 above, and further in view of Kossiakoff.

78. Marmelstein does not teach simulating the flowchart developed using the program code generation tool. Kossiakoff does teach that the circuit diagram is converted into computer code and tested with sample inputs and outputs. The goal is to

correct errors or omissions by the designer. (Kossiakoff, column 4, line 66 – column 5, line 5). It is deemed that this constitutes a simulation of the modeled system. Both Marmelstein and Kossiakoff disclose inventions that allow a user to graphically create a diagram which is subsequently used to generate programming language code. The ability to simulate the graphical program as disclosed by Kossiakoff could have been implemented in the invention of Marmelstein with the addition of a final step of executing the generated program code with sample inputs and reporting the outputs to the user. It would have been obvious to a person of ordinary skill in the art to include features that aid the user to improve the quality of the generated code as this is a goal of both inventions (Marmelstein, column 3, lines 16-40; Kossiakoff, column 14, lines 36-46; column 5, lines 1-5).

79. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kossiakoff as applied to claim 16 above, and further in view of Marmelstein.

80. Kossiakoff does not teach generating the diagram description in a batch mode, however Marmelstein teaches using the input of a complete ADA program package file and generating a complete diagram for that file (Marmelstein, column 3, lines 9-13; column 18, lines 36-49; column 27, lines 8-13). Such a feature could be implemented in the invention of Kossiakoff in an identical fashion without interfering with any existing functionality or goal of the invention. It would have been obvious to a person of ordinary skill in the art to recognize the advantage of generating a diagram from textual program code (Marmelstein, column 27, lines 15-23) and to incorporate said advantage in the invention of Kossiakoff to produce a tool with greater functionality.

81. Claims 30-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Marmelstein in view of Fukumochi et al.

82. Regarding Claim 30, Marmelstein teaches a computer program product which

receives input from the user that specifying a program in the form of a graphical flowchart (column 2, lines 58-63),

the elements of the flowchart labeled using a natural language (figure 5, elements 522, 524, 526; figure 6, elements 622, 623),

the tool represents objects using rectangles (column 17, lines 52-65; figure 5),

the tool represents operations associated with the objects using circles (column 17, lines 59-65; figure 5; column 22, lines 25-41; figure 6),

the tool presents a textual representation of the flowchart (column 2, line 63 - column 3, line 4), and

the tool produces a program code textual description of the flowchart (column 2, lines 59-63; figures 15-25; column 11, lines 10-36).

83. Fukumochi et al. teaches a computer program which translates text from one natural language to a second natural language (column 3, lines 46-51). The combination of the translation device disclosed by Fukumochi et al. with the programming code generation tool disclosed by Marmelstein could be realized as a feature initiated by user command to translate the labels of the various graphical elements. Doing so would facilitate the generation of a new textual program code wherein the labels in the second natural language would be represented in the program

code. It is deemed that the advantages of a translating the labels of a flowchart representing a proposed program design into a second language are inherent in view of the various natural languages spoken by software developers. It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to incorporate the translation ability of Fukumochi et al. with the automatic diagramming and code generation ability of Marmelstein to improve the usability of the code generation tool.

84. Regarding Claim 31, the limitations of Claim 30 are rejected as above, further Marmelstein teaches a program code generation tool wherein the tool models objects as rectangles in the APEX Editor (column 17, lines 52-65; figure 5) and operations associated with the objects are independently represented as circles in the REM-Net Editor (column 17, lines 59-65; figure 5; column 22, lines 25-41; figure 6).

85. Regarding Claim 32, the limitations of Claim 31 are rejected as above, further Marmelstein teaches a program code generation tool wherein the tool produces an ADA programming language textual description of a flowchart (column 2, lines 59-63; figures 15-25; column 11, lines 10-36). The use of context-free grammars to define programming language syntax is well known; indeed this was the first application of context-free grammars (See Sebesta, pages 109-110). A context-free grammar for ADA is known (See The Ada 95 Reference Manual, Annex P). Thus generating an ADA programming language textual description is deemed equivalent to generating a context-free grammar expression.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (703) 305-0542 but will change to (571) 272-3713 starting in October 2004. The examiner can normally be reached on 8am-4pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin J Teska can be reached on (703) 305-9704, or (571) 272-3716 starting in October 2004. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

jsp

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER